Claims:

1. A compression/decompression algorithm to compress/decompress random/not random data and comprising the following processes:

a) compression process that performs compression mainly based on the bits of a sorting algorithm and takes advantage of the fact that a sorting's algorithm bits may be smaller than the combinations it may represent and so it doesn't necessary require the use of DPCM and/or doesn't necessary require a dictionary with DPCM compression and/or variants

and/or

b) decompression process that reverses the process in 1a and thus uses the 'compressed' bits of the algorithm to sort/un-sort in order to form the uncompressed, original bit-string

2. Any variation of claim 1 that doesn't necessary require any other type of dictionary-compression algorithm with any sort of encoding techniques including, but not limited to: Lempel-Zif, Huffman coding, BWT (Burrows-Wheeler Transform), PCM (Pulse Coded Modulation), DPCM (Differential Pulse Coded Modulation), or any code modulation formats, Discrete Cosine Transform (DCT), Wavelet Transform (WT), Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), hashing techniques or any form of Integer Compression or any other form of lossless data compression/encoding

3. Any variation of claim 1 that improves its performance resulting in only faster (or lower) compression/decompression speeds or resulting in only higher (or lower) compression rates/ratios.

4. Any variation of claim 1 that performs compression mainly based on the bits of a sorting algorithm and that takes advantage of the fact that a sorting algorithm's bits may be smaller than n! combinations that it may represent (for instance; 8! = 8*7*6*5*4*3*2*1 = 40320 combinations, or 15.3 bits and the sorting algorithm may only require 13 bits, so we win 2 bits for this sample)

5. Any variation of claim 1 using any other kind of sorting/un-sorting algorithm including all sorting/unsorting algorithms based on encoding, hashing and of any compressed type

6. Any variation of claim 1 using other compression algorithms in order to further compress the data, such as the BWT, hashing techniques or any encoding techniques or such as those mentioned in claim 2.

7. Any variation of claim 1 resulting in compression of other size(s) of blocks (other than described in the description) or using any statistical method to win bits (other than described in the description), or converting the bits to any cardinal number and back (other than described in the description)

8. Any variation of claim 1 by compressing any item size, for example, a 90-bit block size may contain 6 items of 15-bit so here we use a 15-bit item size, so any item-size is possible, even a 3-bit size or a 1024-bit size is possible

9. Any variation of claim 1 whether the data to be compressed/decompressed as stated in claim 1 is truly random data or not.

10. Any variation of claim 1 to compress/decompress data more than once, over multiple passes in order to increase the compression rate/ratio, for example 100 passes may compress a file of 1Gbyte to 1Mbyte

11. Any variation of claim 1 to compress/decompress data more than once, with an n-th selection in order to continually further compress the data in another way

12. Any variation of claim 1 in combining the combinations of several conversions; for instance, we could combine many of our our 8 numbers of 3-bits and also in order to come closer to our 2^X number, for instance; these 3 combinations: 10423675, 61432570 and 21037654, each represent 15.3 bits, in total = 15.3*3 = 45.9 bits and when combined, they may represent 46 bits, or 2^46 possible combinations

13. Any variation of claim 1 to compress/decompress data in a realtime / streaming way including any hardware and/or software required to perform it

14. Any variation of claim 1 by using a separate bit-table or a pre-processing table that may be required by the algorithm in order to improve the statistical change of winning bits, an example of a bit-table is: to include a bit per compressed 15-bit item saying 0 = needs shell sort increments of type A, or 1 = needs shell sort increments type B, another example is: 0 = don't compress 15-bit item, 1 = compress 15-bit item

15. Any variation of claim 1 by making use of any 'statistics' (for instance; winning 10 bits and using a pointer of 9 bits to tell its position in the bit-string and thus win one bit in the process)

16. Any variation of claim 1 by making use of the modular (mod) function for direct calculations and/or any-table-like structure to convert/compress/decompress our original bitstring to our sorting-bitstring and back

17. Any variation of claim 1 to build the compression/decompression steps with unique or not-unique digits, other than described in the description, for instance 10423675 has 8 unique digits, 10423474, has 4 coming up 3 times and the numbers 5 and 6 are missing

18. Any variation of claim 1 to unsort any digits/numbers, other than described in the description, in order to convert/compress/decompress our original bitstring to our sorting-bitstring and back, for instance; for 8 numbers of 3-bit (0-7): 01234567 becomes 10423675, or, for 4 number of 2-bit (0-3); 0123 becomes 3120

19. Any variation or application of claims 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,10 ,11 ,12 ,13, 14, 15, 16, 17, 18 by using any part of the algorithm and making it part of another algorithm and so as to achieve the same goal but in another way

20. Any application of claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,19 using this technology in any industry including but not limited to software industry, data storage industry, digital entertainment industry, data transmission industry, telecommunications industry

21. Any application of claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 to use this technology including but not limited for areas such as file compression in general, compressed solid state disks, data transmission, video compression (like MPEG, AVI etc), network/modem compression (Radio/Telephone/DSL/Cable/Satellite etc), music compression (like MP3 etc), graphic compression (like JPG etc), database compression (like SQL Server etc), speech compression (like cell phones etc), communications (like Cell Phone, PDA, Pocket PC, broadcast TV, Interactive TV, cell phone video conferencing, Fax Devices, Handhelds etc), digital entertainment (like DVD, CD, etc), any form of data encryption, to compress real-

time/streaming (random) data, for use to compress video / images, or for use in voice recorders, memory sticks, data banks, external data storage devices, all types of memory cards and external storage media, like Compact Flash, MicroDrive, SmartMedia, SD, MMC, MemoryStick (-Pro), Datapak, Dataflash, Smart Media, USB Memory, PC Card Hard Drives etc.

22. Any application of claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 using this technology as software based or hardware based, included but not limited to chips such as FPGA, ASIC, CPLD, EPLD, PLD

23. Any application of claims 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 using this technology in combination with any other data compression algorithms such as those used by GIF, JPEG, ZIP, MPEG, MP3 and others